

FIG. 1

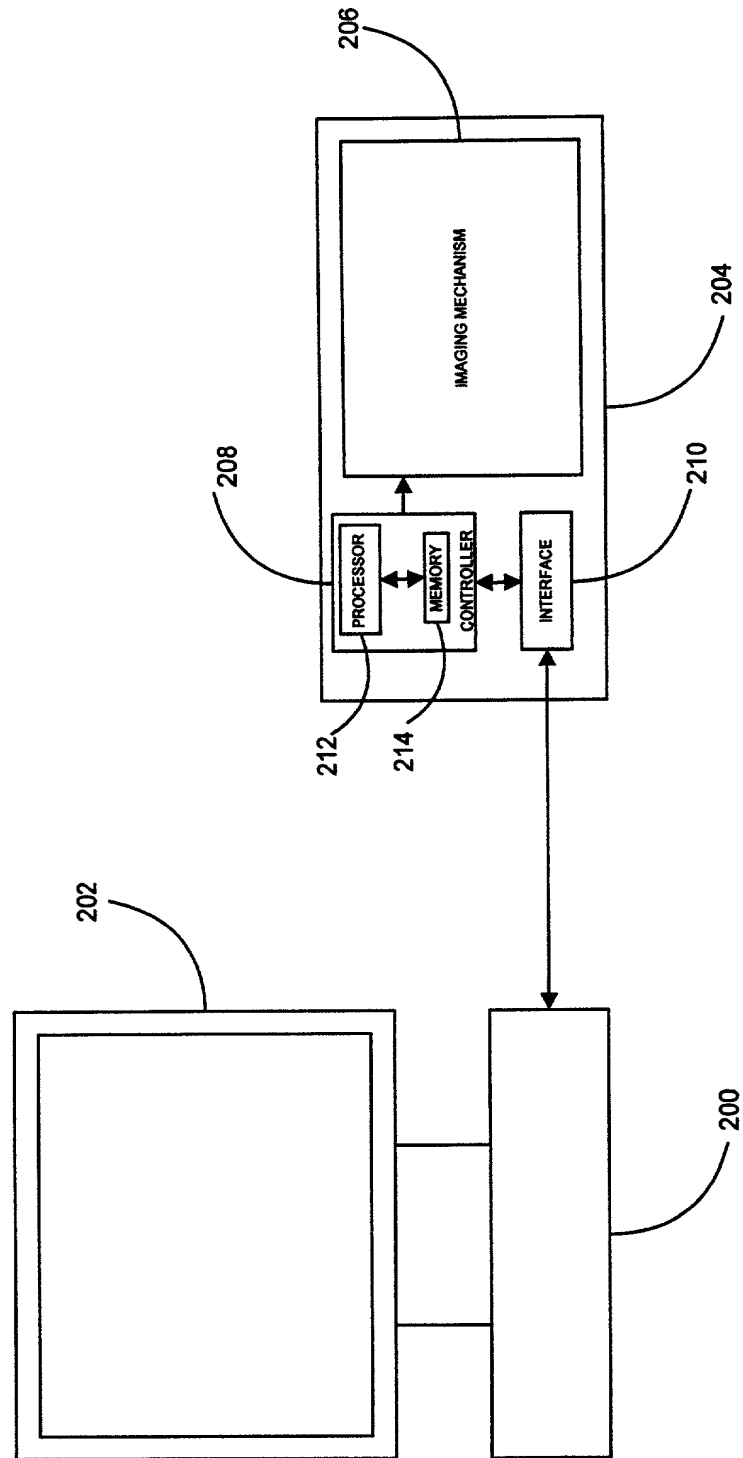


FIG. 2

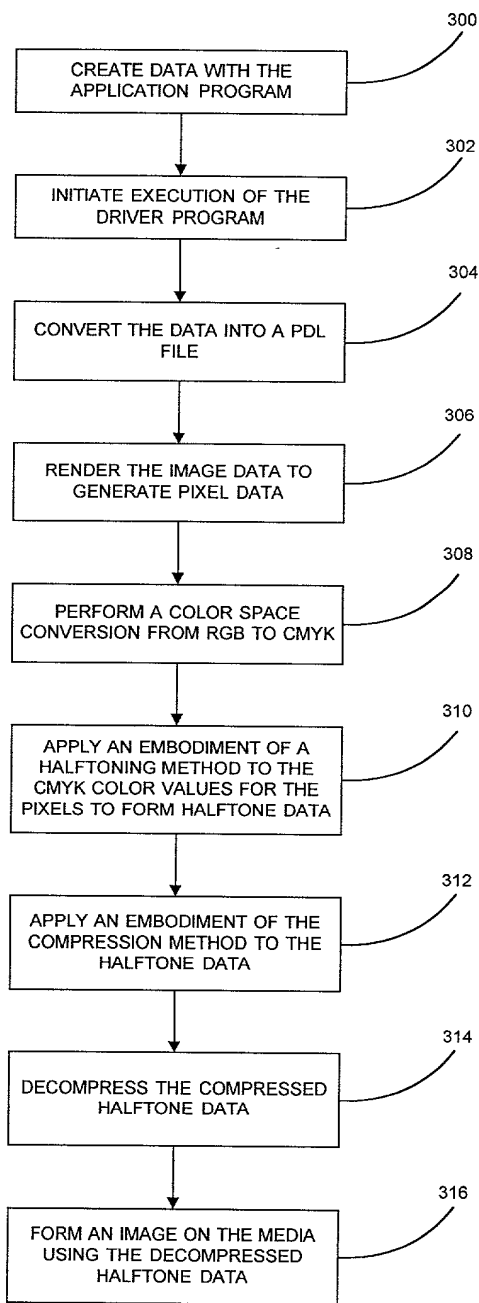
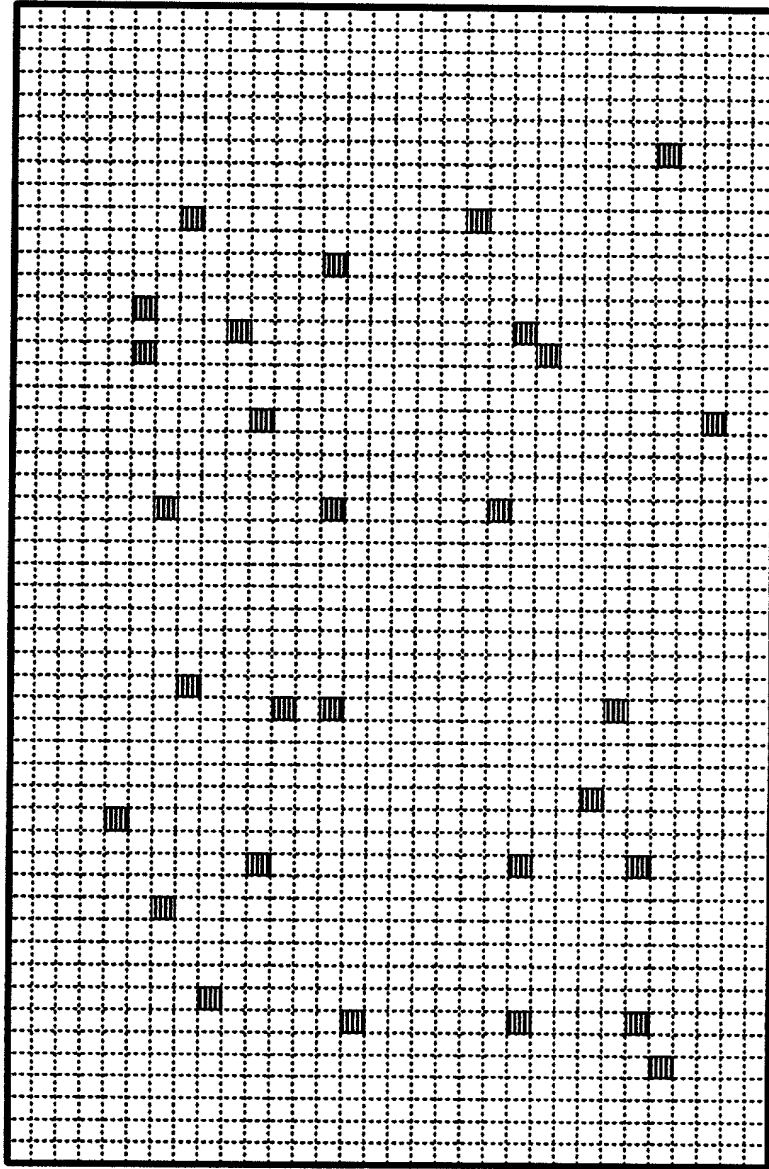


FIG. 3

FIG. 20

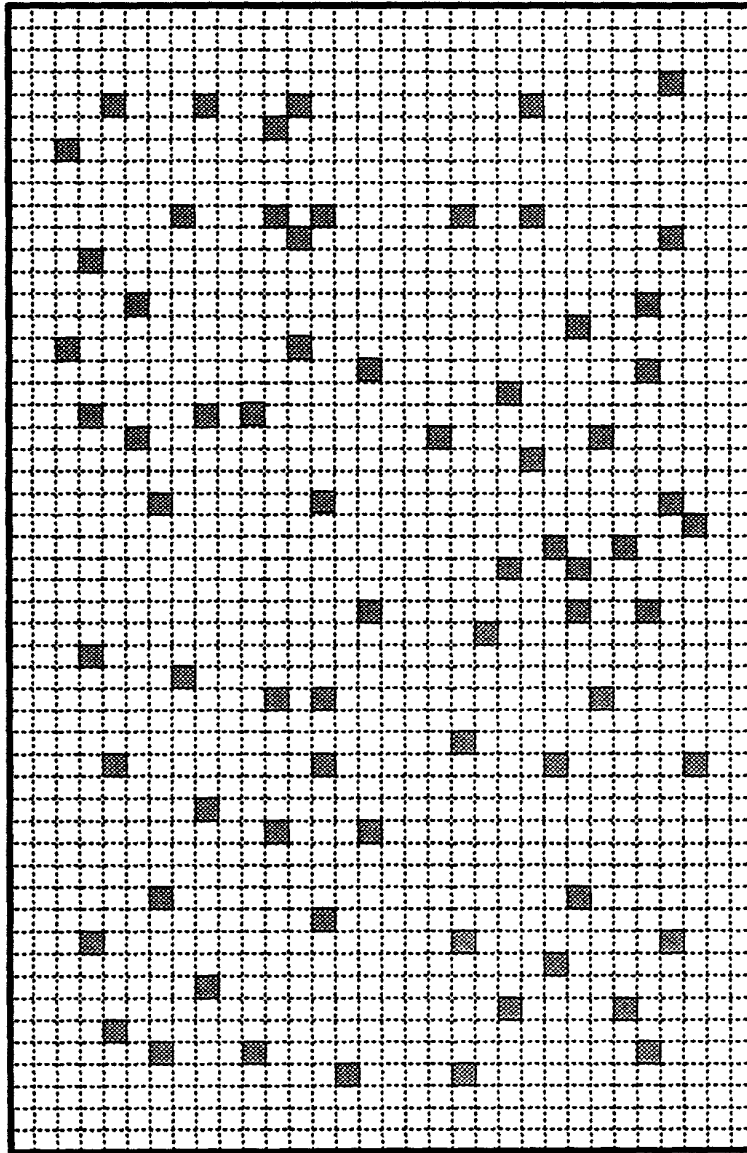


400
CYAN

NOT CYAN

FIGURE 4

FIG. 5

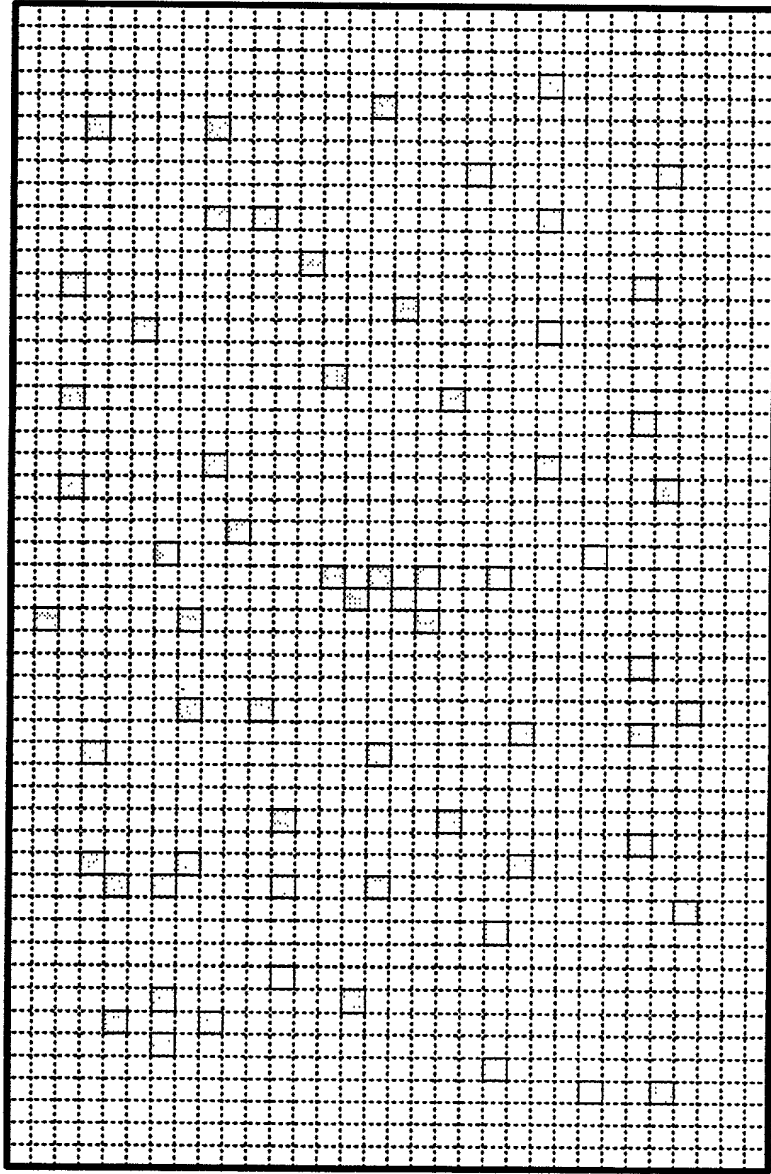


500
MAGENTA

NOT MAGENTA

FIGURE 5

FIG. 6



600

□ YELLOW

□ NOT YELLOW

FIGURE 6



FIGURE 7

FOI E.O. 14176

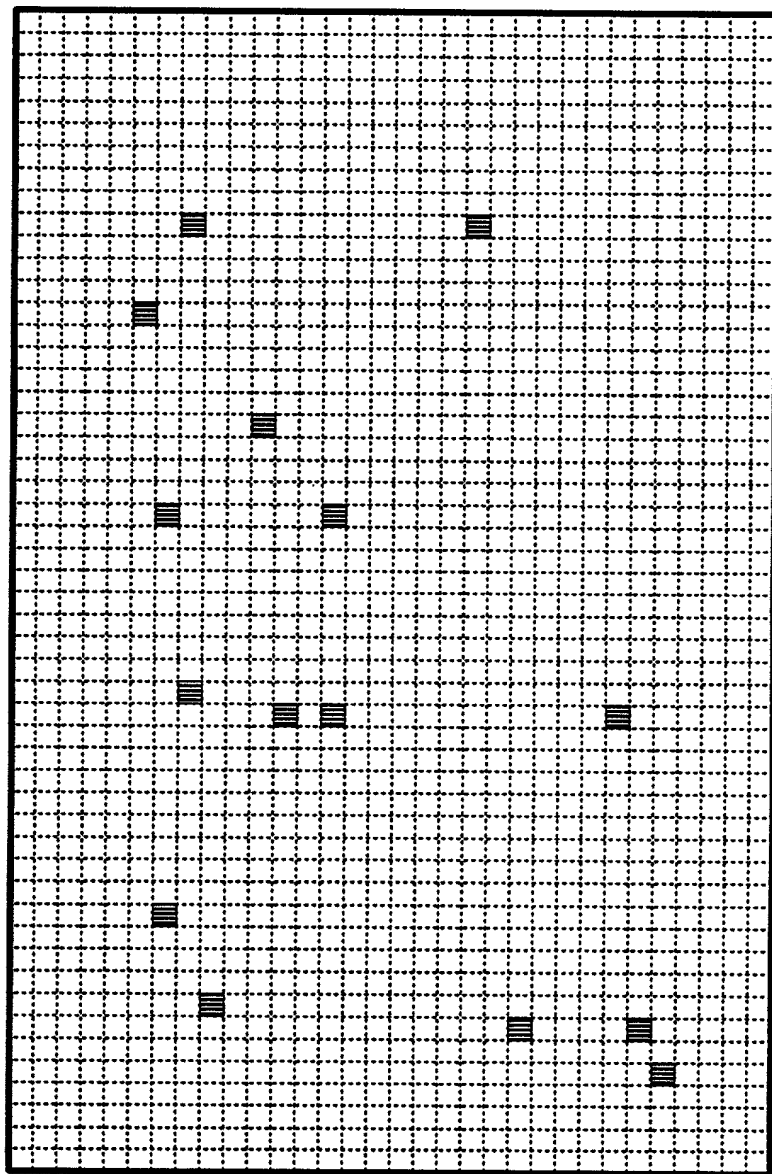
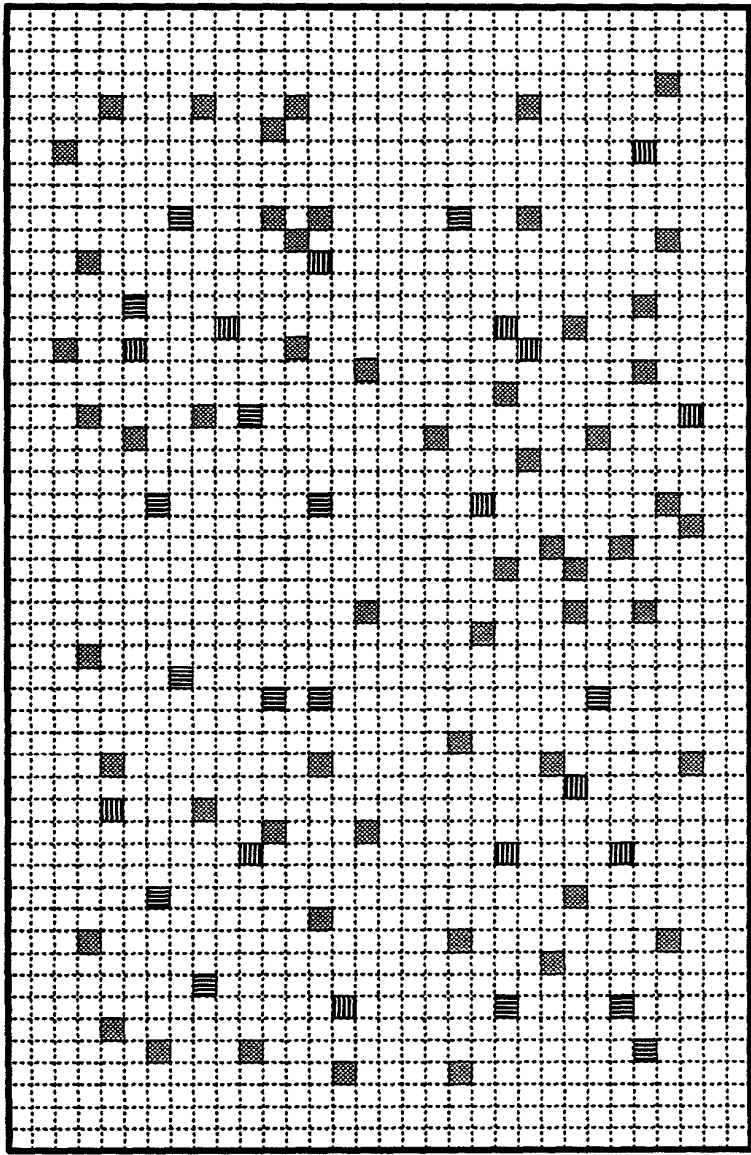


FIGURE 8

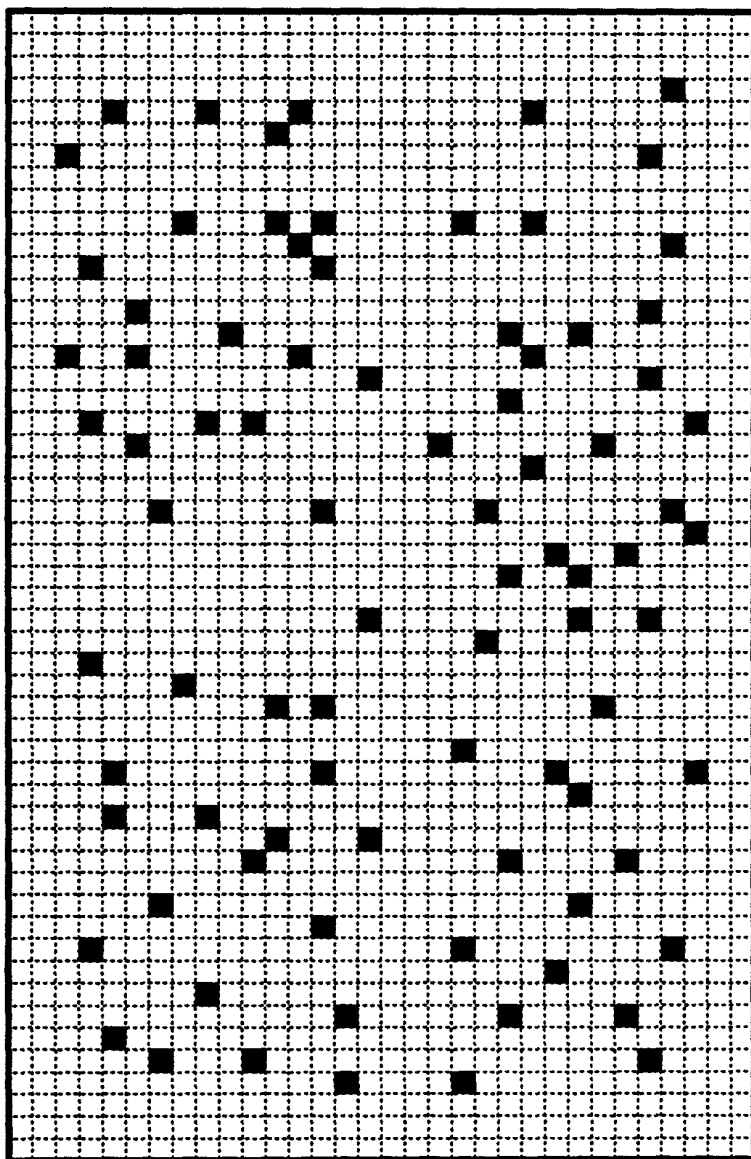
FIG. 20" 24951660



CYAN MAGENTA BLUE NOT CYAN OR MAGENTA

FIGURE 9

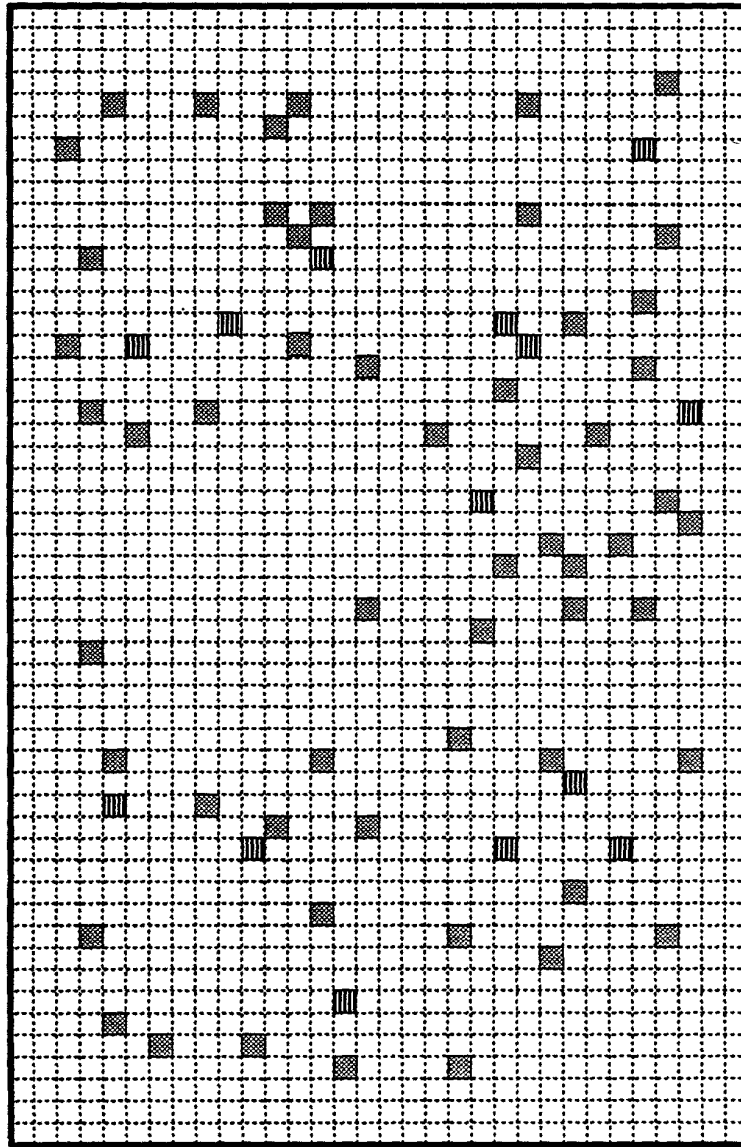
FIG. 10



■ NOT WHITE □ NOT CYAN OR MAGENTA (WHITE)

FIGURE 10

FIG. 11

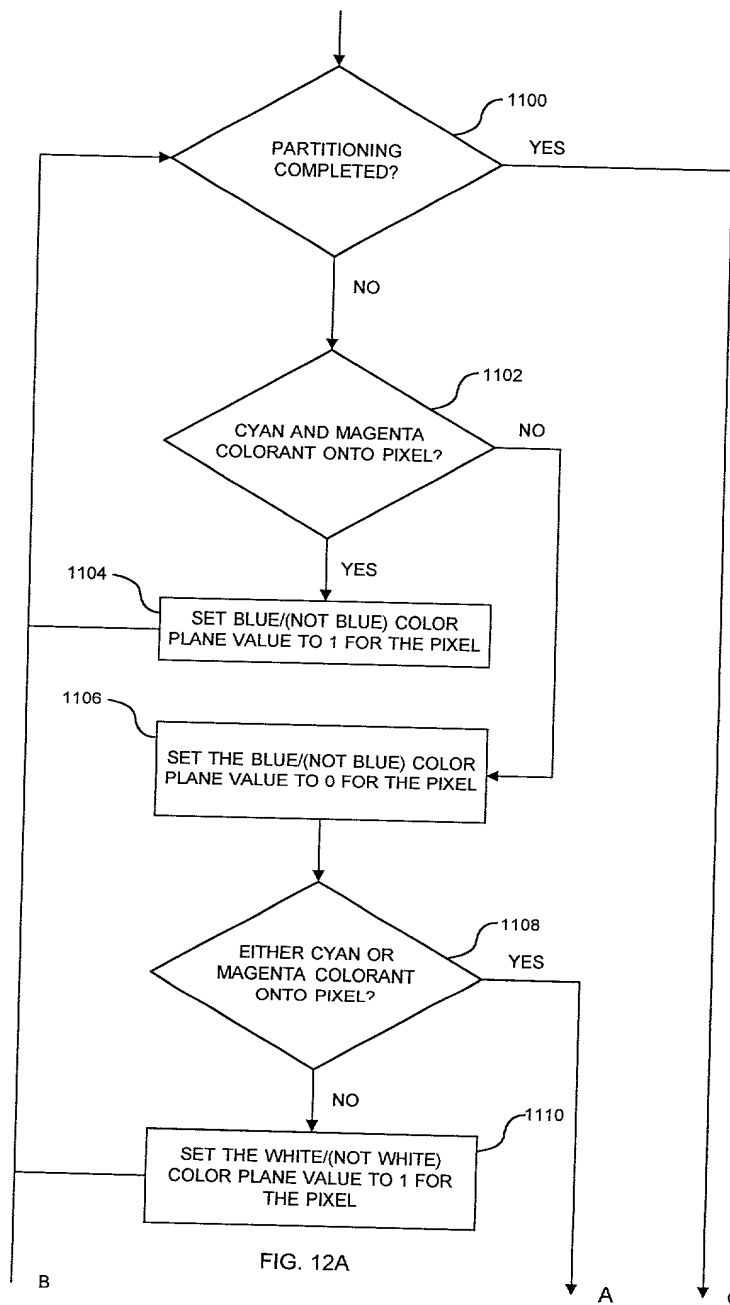


400 CYAN

500 MAGENTA

NOT CYAN OR MAGENTA (WHITE)

FIGURE 11



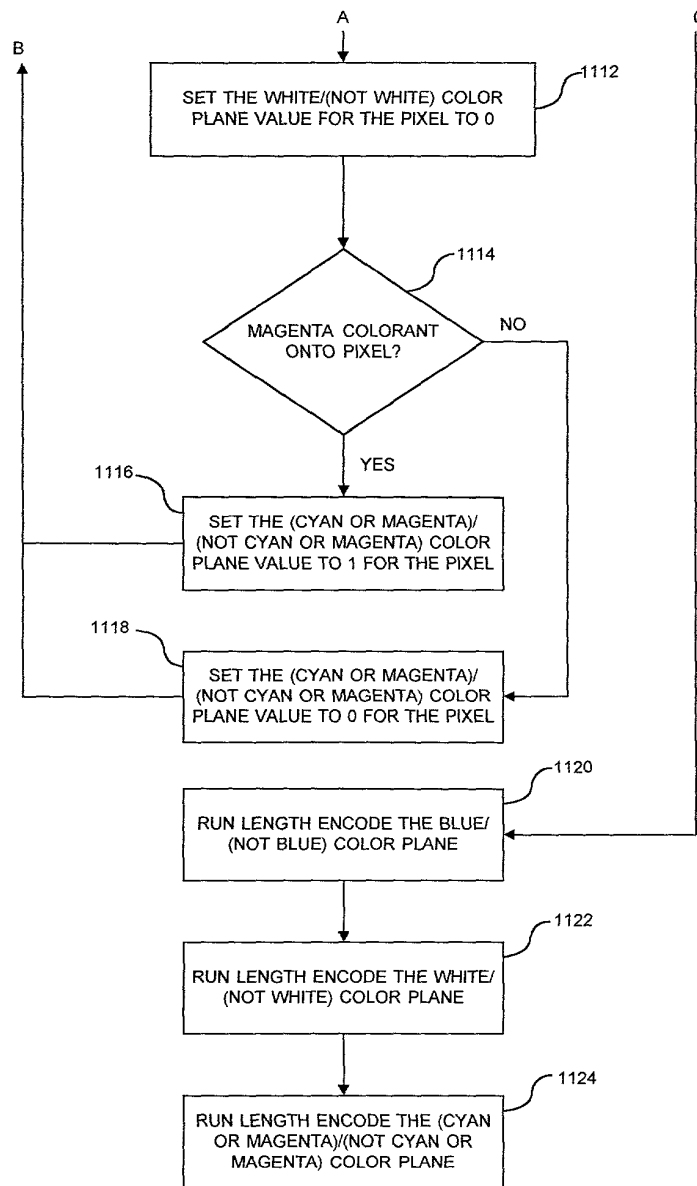


FIG. 12B

09919647-073101

```
for each pixel at i,j, in the halftoned image
{
    // Check C and M together to fill the Blue bit plane
    if m(i,j) == 1 AND c(i,j) == 1
    {
        b_plane(i,j) = 1;    // add a 1 to the blue plane
    }
    else    // not blue, must add entry to white plane
    {
        b_plane(i,j) = 0;    // set blue plane to 0

        if m(i,j) == 1 OR c(i,j) == 1    // if NOT white
        {
            white_plane(w_curr_location) = 0;
            if m(i,j) == 1
                C/M_plane(cm_curr_location) = 1;
            else
                C/M_plane(cm_curr_location) = 0;

            Increment w_curr_location, cm_curr_location;
        }
        else    // its white so add a 1 to the white plane
        {
            white_plane(w_curr_location) = 0;
            Increment w_curr_location;
        }
    }

} // end else blue

} // end for each pixel in ht image

RLE_encode(B_plane);
RLE_encode(W_plane);
RLE_encode(C/M_plane);
```

FIG. 13

UNCOMPRESSED HALFTONE FILE SIZE	COMPRESSED HALFTONE FILE SIZE
3.6 MEGA BYTE	2.6 MEGA BYTE
1.7 MEGA BYTE	1.0 MEGA BYTE
3.6 MEGA BYTE	2.2 MEGA BYTE
3.6 MEGA BYTE	2.1 MEGA BYTE

FIG. 14

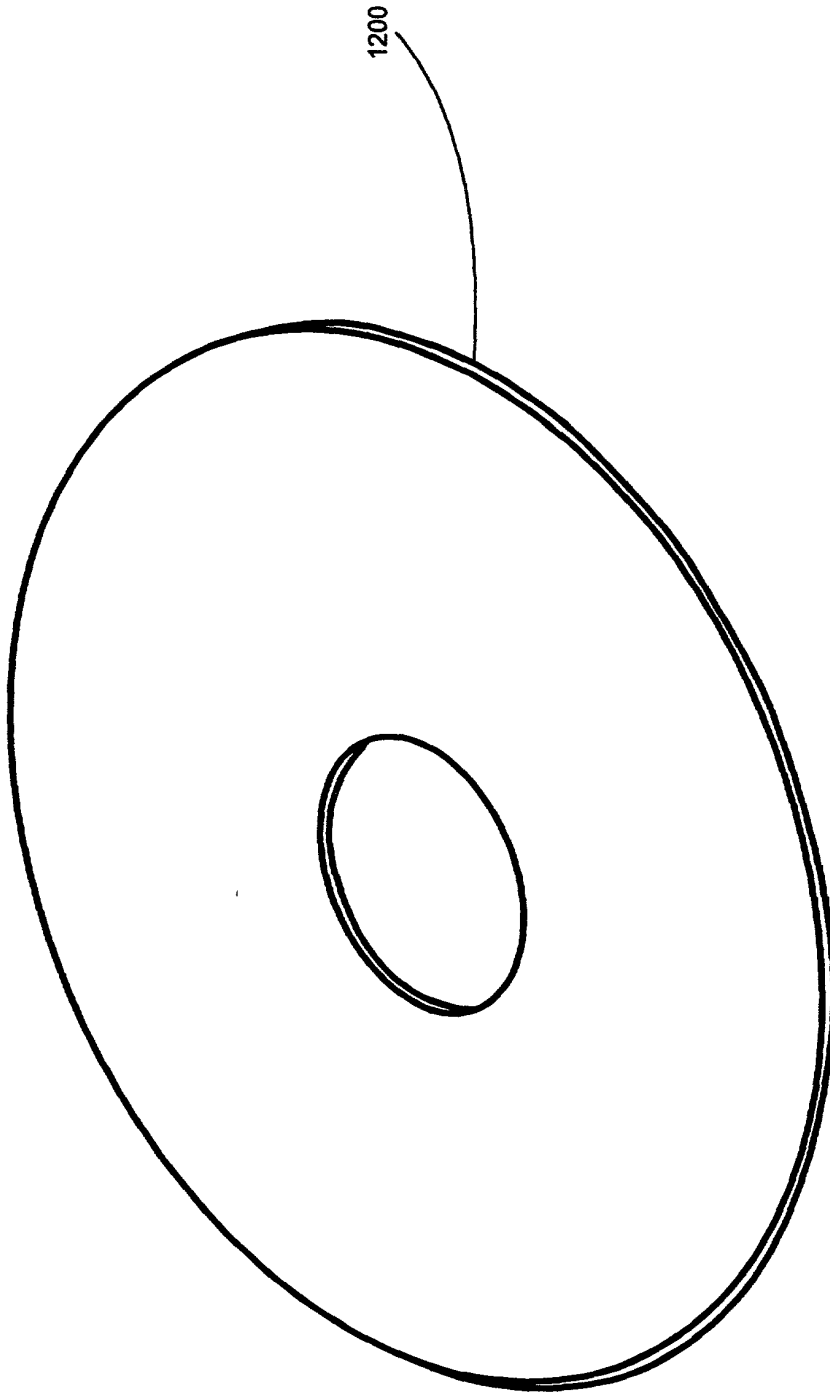


FIG. 15

FIG. 15